Accred Qual Assur (2016) 21:377–380
DOI 10.1007/s00769-016-1228-6

DISCUSSION FORUM

Further on creating normal density functions in Microsoft® Excel

Niklas Bark[1] · **Anders Kallner**[1]

Received: 28 April 2016 / Accepted: 12 July 2016 / Published online: 26 July 2016

Abstract Simulations play an increasing role in metrology and analytical chemistry, particularly the use of simulated normal distributions. Microsoft Excel is available to almost everybody and normal distributions can be simulated using either an innate function or other algorithms. We explore the success of five different algorithms to simulate normal distributions and compare the outcome with a simulation coded in R. The normality of 10^6 data points simulated by the chosen algorithms was tested by different recognized statistical procedures and Q–Q plots. They all failed the statistical procedures, but evaluating the Q–Q plots revealed different types of deviations. It also showed that the deviations, although statistically significant, most likely do not have any practical relevance in laboratory work.

Keywords Simulation of NDF · Normal distribution · Q–Q plot · Normality R-codes

Introduction

Microsoft Excel is a tool which is commonly used for a multitude of calculates. Although initially developed for administrative and business purposes, it is a useful tool also in laboratory work. Typical for the spreadsheet approach, there are often several different routes possible to solve the same problem or task. We have studied some characteristics of six developed algorithms for simulation of a normal (Gaussian) distribution [1, 2]. The study was based on the averages of ten simulations of 10^6 numbers each. The algorithms gave slightly different distribution widths (maximum–minimum result), and some other calculated characteristics. The differences between the recalculated averages and standard deviations were miniscule and agreed with the defined quantities. The tested distributions could be summarized in three different groups, one with algorithms based on additions and subtractions of the term RAND() and derived from a rectangular distribution, another based on the Excel functions NORMDIST, NORMSINV and NORM.S.INV and the RAND() function as "probability." The third group was the Excel Add-in normal function, whose calculation principles are unknown.

Results were presented in a comprehensive table and a series of graphs. The latter were based on histograms, i.e., summaries of 10^6 observations collected in bins. To create the histograms, we used the FREQUENCY function which summarizes the number of observations in the bin and presents them as referring to the upper limit of the respective bin. With equally sized bins and a random distribution of results, the location of the calculated frequency would be better represented by the average of the two bin limits. Our choice therefore distorted the graphs to give an illusion of a right-shift compared to the calculated normal density function even if the latter was calculated from the same bin limits. This virtual "right-shift" corresponds to half the bin width. A slight right-shift of the distribution was, however, also noticeable in the calculated quantity values, e.g., percentiles calculated from the obtained distributions did not correspond to the expected z values (results relative to the standard

Papers published in this section do not necessarily reflect the opinion of the Editors, the Editorial Board and the Publisher.
A critical and constructive debate in the Discussion Forum or a Letter to the Editor is strongly encouraged.

✉ Anders Kallner
Anders.kallner@ki.se

[1] Department of Clinical Chemistry, Karolinska University Hospital, Stockholm, Sweden

deviation). However, for the innate normal function there was a distinct unsymmetrical left-shift, i.e., the simulated distribution was left-skewed.

In the present study, we compared the five Excel simulated density functions with a function simulated by R [3] and test the normality of the generated density functions using Q–Q plots.

Materials and methods

The algorithms described in the previous report (A, B, C, D and E) [1] were used.

The used algorithms were:

(A) AVE + SD × (RAND() + RAND() + RAND() + RAND() + RAND() + RAND() + RAND() + RAND() + RAND() + RAND() + RAND()–6)

(B) AVE + SD × (RAND() − RAND() + RAND() − RAND() + RAND() − RAND() + RAND() − RAND() + RAND() − RAND())

(C) NORM.INV(RAND(),AVE,SD)

(D) AVE + SD × NORM.S.INV(RAND())

(E) AVE + SD × NORMSINV(RAND())

For the simulation by R, the code in Table 1 was used. One million numbers (10^6) were generated by each algorithm using an average (AVE) of 10 and a standard deviation (SD) of 0.5 for positioning and distribution width, respectively. The Excel Add-in was not included in the present study since it is limited to about 37 000 observations.

The R-simulated dataset of 10^6 normally distributed values was assumed as a reference and assumed free from any ambiguity. The datasets were then compared using Q–Q plots [4] by which a distribution can be compared with any other defined distribution. The average of the slopes and intercepts of the regressions of the Excel-generated datasets on the R-generated data and on data from the average of ten simulations by algorithm C were calculated (Fig. 1). We thus obtained two sets of five slopes and five intercepts. These were subjected to comparison using an ANOVA. The average, standard deviation and 95 % confidence interval (CI) were calculated for each set of slope and intercept.

Normality tests of the simulated density functions were also performed using Prism (GraphPad, San Diego, CA, USA). which provides the D'Agostino-Pearson omnibus and the Kolmogorov–Smirnov normality tests. The Shapiro–Wilks test, which is also available in Prism, only allows up to about 5 000 observations.

Results

The agreement between the distributions of two datasets can be tested by comparing their quantiles. If identical the quantiles will be found along a straight line with a slope of 45° and an intercept of zero (provided the scales of the axes are the same). In a graphical presentation, the proximity to the equal line and the regression between the tested distributions can be evaluated. This is the principle of the Q–Q plot. If one of the datasets is normally distributed, the normality of the comparative dataset can thus be estimated.

Table 1 R-code to create a simulated normal density function curve (1) and the code for realizing algorithms A and B (2)

(1) *# generate a set of 10^6 normally distributed numbers*

AVE = 10 SD = 0.5

set.seed(2) random numbers = rnorm(1000000,10,0.5)

(2) # *Algorithms A and B*

n = 1000000, AVE = 10, SD = 0.5

set.seed(2)

#*Create 12 columns with 10^6 random numbers (default 0 to 1)*

df = data.frame(uni1 = runif(n), uni2 = runif(n), uni3 = runif(n), uni4 = runif(n),

uni5 = runif(n), uni6 = runif(n), uni7 = runif(n), uni8 = runif(n), uni9 = runif(n),

uni10 = runif(n), uni11 = runif(n), uni12 = runif(n))

#*Algorithm A: Add all and subtract "6"; transfer to a new column.*

Average = 10, S(x) = 0.5

df$MOD_A = 10 + 0.5 *

(df$uni1 + df$uni2 + df$uni3 + df$uni4 + df$uni5 + df$uni6 + df$uni7 + df$uni8 + df$uni9 + df$uni10 + df$uni11 + df$uni12 − 6)

#*Algorithm B Add and subtract every second random number, transfer to a new column*

df$MOD_B = 10 + 0.5 *

(df$uni1 − df$uni2 + df$uni3 − df$uni4 + df$uni5 − df$uni6 + df$uni7 − df$uni8 + df$uni9 − df$uni10 + df$uni11 − df$uni12)

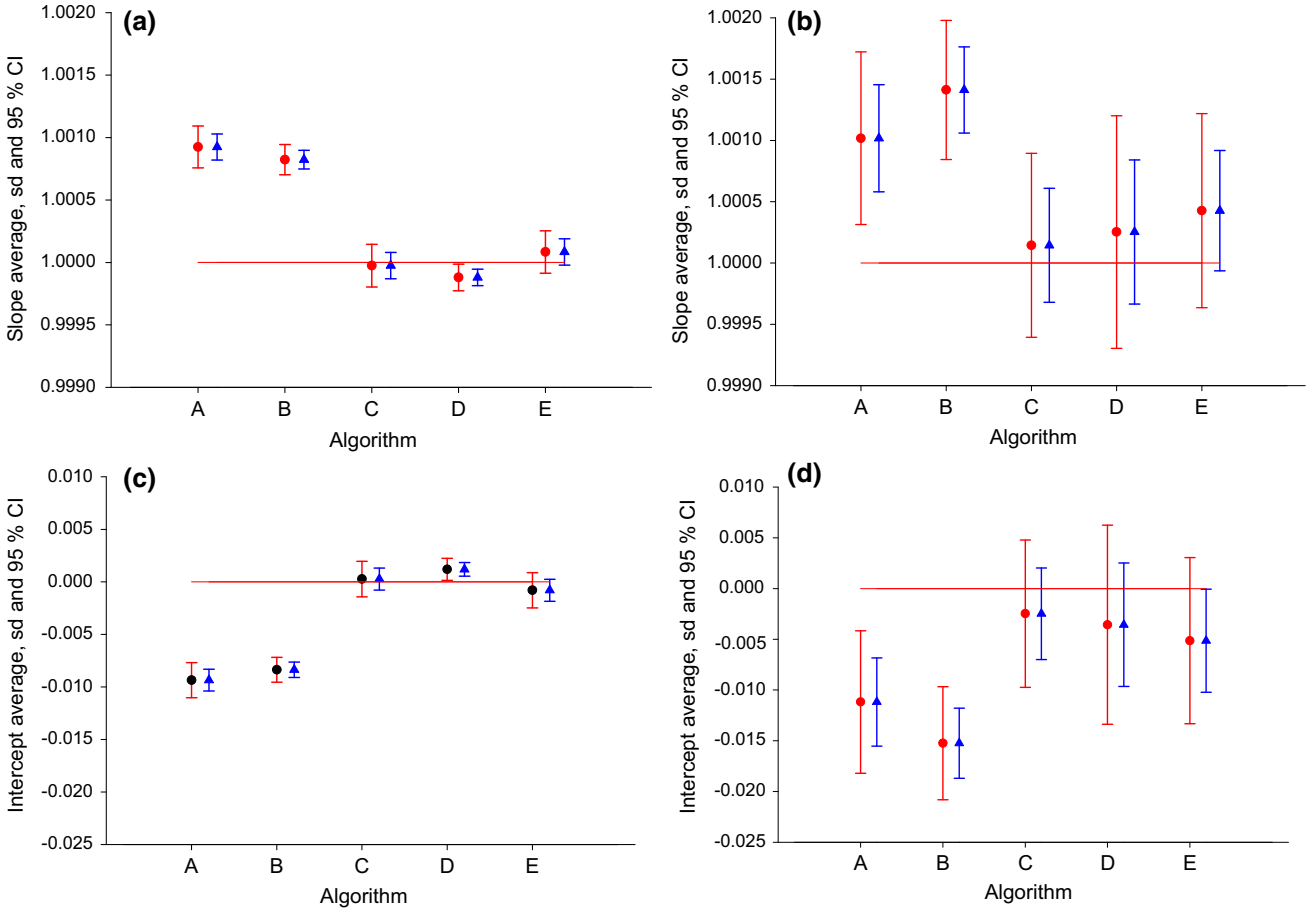

Fig. 1 Evaluation of Q–Q plots: Averages of slopes (**a**, **b**) and intercepts (**c**, **d**) of the regression line between the simulated (10 times) frequency distribution algorithms (A through E) and the algorithm C (**a**, **c**) and the R-generated normal frequency distribution, (**b**, **d**). The *horizontal lines* mark the target values of an equal line. *Circles* and *error bars* indicate the average and standard deviation, *triangles* and their *error bars* the average and 95 % CI

The correlation coefficients of the regression between the tested density functions and the reference in the Q–Q plots were invariably 1.0000.

The ANOVA indicated significant differences ($p < 0.05$) between the slopes and between the intercepts of the Q–Q plots of the simulated datasets and their references (algorithm C and R-coded). Thus, the slopes were higher in the comparisons between simulations by algorithms A and B and the reference than between algorithms C, D and E and the reference (Fig. 1). The relations were the opposite for the intercepts, i.e., lower for A and B than C, D and E. A similar pattern was obtained if the simulated data were compared with a normal distribution represented by a single simulation using the algorithm C as reference (Fig. 1). As expected, the slope and intercept for the algorithm C were inseparable from 1 and 0 ($\alpha = 0.05$), respectively. The uncertainty of the comparison with the R-generated data was larger than that obtained in the comparison with that of algorithm C.

The density functions based on 10^6 observations and created by algorithms A and B did not pass the computed normality tests, whereas the others did. The algorithms A and B, when coded in R, also did not pass the normality tests. With smaller sample sizes, about 18 000 observations, the B algorithm but not the A algorithm did pass the Kolmogorov–Smirnov test. The data of the A and B algorithms did not pass the D'Agostino-Pearson omnibus normality tests until sample sizes of 1 000 and 10 000 observations, respectively.

Discussion

A previous study [1] indicated a "right-shift" of simulated density functions in relation to the calculated Gauss function. In an erratum [5], it was suggested that this is partly an effect of how the used function in Excel reports the frequency. However, other statistics, which are reported in

Table 1 in ref [1], gave some support to the conclusion. A noteworthy feature of the algorithms A and B is a slightly smaller interval between minimal and maximal values; about 10 % in comparison with the algorithms C, D and E.

In the present study, we explore some other features of the simulation algorithms and compare the outcome with a density function simulated using R-coded data for reference. We chose R as a reference since its statistical functions seem less criticized than the ones in Excel. The comparisons between Excel and R-generated simulations show the same pattern but the uncertainty in the comparisons of Q–Q plots is larger with the R-generated simulations. This does not discriminate the R-simulations but may rather indicate a difference of the pseudo-random generation. The algorithms A and B were deduced in our previous study, and all the algorithms critically include a pseudo-random generated function.

We speculate that the finding that samples comprising 10^6 data points did not pass the normality tests at a large number of observations can be attributed to a higher statistical "power," i.e., sensitivity to deviations from normality. This is supported by the finding that datasets with fewer observations, sampled from the original 10^6 data point simulations, eventually passed the tests. There was a remarkable difference between data generated by algorithms A and B compared to those generated by C, D and E.

In the Q–Q plot, the slopes of simulations by algorithms A and B are about 1 % above the slope of the equal line (slope 1 and intercept 0) although this seems "compensated" by a negative intercept (Fig. 1). The difference may be statistically significant, but is not likely to be practically relevant.

References

1. Kallner A (2016) A study of simulated normal probability functions using Microsoft Excel. Accred Qual Assur 21:271–276. doi:10.1007/s00769-016-1200-5
2. Kallner A (2015) Microsoft Excel 2010 offers an improved random number generator allowing efficient simulation in chemical laboratory studies. Clin Chim Acta 438:210–211
3. The R Project for Statistical Computing www.r-project.org. Accessed July 2016
4. NIST/SEMATECH e-Handbook of Statistical Methods, http://www.itl.nist.gov/div898/handbook/1.3.3.24. Accessed July 2016
5. Kallner A (2016) Erratum to: a study of simulated normal probability functions 3 using Microsoft Excel. Accred Qual Assur 21:277. doi:10.1007/s00769-016-1224-x